AFRL-IF-RS-TM-2001-9
In-House Technical Memorandum
January 2002

FEATURE EVALUATION TOOLS

Brian Costello

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TM-2001-9 has been reviewed and is approved for publication.

APPROVED:

RICHARD J. SIMARD
Acting Chief, Multi-Sensor Exploitation Branch
Information & Intelligence Exploitation Division
Information Directorate

FOR THE DIRECTOR:

JOSEPH CAMERA
Chief, Information & Intelligence
Exploitation Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFEC, 32 Brooks Road, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 2002	In-House Tech Memo, Jun 01 - Aug 01

4. TITLE AND SUBTITLE

FEATURE EVALUATION TOOLS

5. FUNDING NUMBERS

PE: 62702F
PR: 459E
TA: PR
WU: OJ

6. AUTHOR(S)

Brian Costello

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFEC
32 Brooks Road
Rome, NY 13441-4114

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-IF-RS-TM-2001-9

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFEC
32 Brooks Road
Rome, NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFR:L-IF-RS-TM-2001-9

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Dr. Andrew Noga/IFEC/315-330-2270
Brian Costello is a participant in AFRL's summer student engineering employment program.

12a. DISTRIBUTION AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Measurements or transformations of measurements (i.e. features) are often used for distinguishing between items or groups of interest. For example, cepstral features are useful for speaker identification. Given a large set of possible features, the problem is choosing which to use and which not to use in order to optimize speed and performance. A commercially available software package, MATLAB (Mathworks, Inc.) has been used to implement previously developed methods of feature evaluation tools. These tools include the well known Mahalanobis and Battacharyya distances, along with a class inter-to-intra measurement ratio. A synthetic data generator for creating multi-variate Gaussian classes was also implemented to aid in the development of the evaluation tools.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Feature evaluation, statistical pattern recognition, Mahalanobis distance, Battacharyya distance	52
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

Table of Contents

1.0 Introduction

Note: This report is to be viewed in respect to past reports. For further information, see reports [1], [2], & [3].

STATPACK is a statistical pattern recognition tool that utilizes the Matlab (The Mathworks Inc.) language and environment. It was originally developed by S.P.Montana under the Summer Engineering Aide Program from June 1996 to August 1996 [1]. R. M. Floyd then made modifications during the period of September 1996 to December 1996, increasing the program's speed and efficiency [2]. S. P. Montana made further enhancements under the Engineering Aide Program from June 1997 to August 1997 [3].

The original code was written in Matlab version 4.2c, and new code and revisions have been written in Matlab version 5.3. Because both versions are used, thought and effort have gone into making the program completely compatible with both versions. However, compatibility with the new version 6 has not yet been confirmed.

This report covers the changes also made under the Engineering Aide Program from June 2001 to August 2001 by B. P. Costello. These latest changes were made using Matlab version 5.3, and were designed to enhance the utility and effectiveness of STATPACK as a whole. They included appropriate compatibility corrections, as well as new 3-D plotting and feature evaluation functions.

A list of the functions written by S. P. Montana, R. M. Floyd, and B. P. Costello can be found on pages 23 and 24.

Example Plots:

- R. M. Floyd's 1-Dimensional Analysis Plots can be found on pages 26 – 27.
- S. P. Montana's 2-Dimensional Analysis Plots can be found on pages 28 – 29.
- New 3-Dimensional Analysis Plots can be found on pages 30 - 31.
- New Feature Relative Worth Plots can be found on pages 33 - 34.

2.0 Adjustments / Corrections

Because STATPACK was originally developed under Matlab 4.2, some minor plot modifications had to be made when bringing the program over to version 5.3. Other modifications were also necessary to make code written in 5.3 usable in version 4.2. Also, the Nearest Mean Classifier had not been fully debugged, and STATPACK could not detect the absence of a node unless the program was restarted after all of the nodes had been deleted.

2.1 Compatibility Corrections

Because of the continuing use of Matlab version 4.2, effort has gone into making sure that STATPACK is compatible with that version. The main problem was the use of Matlab 5.3 functions that were not defined in version 4.2. These included the functions "mod" and "strcat".

Next, due to differences in default plot settings, colors had to be switched around, and a new option was added. Paragraph 2.2 will go into more depth on this subject.

Other various syntax problems also arose and were dealt with. The result is a program that is now functional for a wider range of users.

2.2 Analysis Plot Changes

Because of the differences between Matlab versions 4.2 and 5.3, certain adaptations had to be made. First, the option to have a black or white background color has been added. The purpose of this is to allow the user to choose their preference. If they are used to using Matlab version 4.2, they will probably prefer the black background, and vice-versa.

Also, the tags that had been plotted in white were changed to gold. This helps to distinguish them from the "hidden" classes; otherwise, they would also blend in to a white background.

The corrected routines sit under the "Analysis" menu, and the "1D-Structure" and "2D-Structure" submenus

2.3 Nearest Mean Classifier

As stated in [3], the Nearest Mean Classifier function had not been fully developed. Work has been done on this area of the program, to correct certain errors. The full-set / self-classify option is now working completely. However, further work still needs to be done to correct other areas of the code.

2.4 No Node Problem

STATPACK was having a problem detecting if there were no selected nodes. The problem occurred when all nodes were removed, and an application was run. Unless STATPACK was exited and restarted, it would not recognize that there were no selected nodes, and would attempt to execute the selected function. That problem has now been corrected.

2.5 Node Deletion Problem

STATPACK was also having another problem with the node deletion process. The deletion function was written such that it would also delete the directory tree for that node in the STATPACK data directory. However, the function was written to use the dos command "deltree". The problem is that the dos prompt no longer recognizes that command. Therefore, the command was changed to "rd /s /q".

2.6 Other Corrections

Other problems arose if the user accidentally clicked on the main screen immediately after selecting an option that began with either a feature selection screen or a wait-bar. First, if the option started off with a feature selection screen and the STATPACK main screen was selected quickly, the feature menu would be created on the main screen, and it would not work. Now, it will not print on the main screen, which prevents it from crashing. Second, if the option began with a

wait-bar and the STATPACK main screen was selected, an axis would be drawn on the main screen, and the function would crash. An axis will still be drawn on the main screen, but the function will continue on, and no real damage is done.

Next, because STATPACK uses new .mat files in the node's directory (that will be described in further detail later), there needed to be a way to clear them out. The problem arose because STATPACK detected that they existed, and then used the existing files. However, if the existing .mat files apply to a past node with the same name, the program would output incorrect data. Now, when a new node is loaded into STATPACK, it will delete all .mat files in an existing node before replacing it.

3.0 New Additions

During the summer of 2001, new functions and new options were added to the STATPACK environment. These include 3-dimensional plotting functions to compliment the already existing 1 and 2-Dimensional plotting functions and feature evaluation metrics with feature relative worth calculators. Also included are a process comparison function, a data set decorrelator, and a data set generator.

3.1 3-D Analysis Functions

As an addition to the already existing 1 and 2-Dimensional Analysis functions, 3-Dimensional Analysis functions have been added. These functions have been adapted from the 2-Dimensional Analysis functions written by S.P.Montana [1], and allow the user to select three axes for the data to be plotted

onto. The true beauty of these functions is the ability to rotate the plots in order to achieve the maximum separation from a single viewpoint. These functions utilize the intrinsic function plot3, as well as the new STATPACK function plot3d. They can be run by selecting their respective menu option under the "Analysis" menu and the "3D-Structure" submenu on the STATPACK main screen. Example plots can be found on pages 30 - 31.

3.1.1 3-D Feature Projection

An adaptation of S2crdv.m, S3crdv.m allows the user to select three features to use, instead of two. The function then plots the data according to the values of the data with respect to each selected feature. An example of a 3-Dimensional Feature Projection can be found at the top of page 30.

3.1.2 3-D Eigenvector Projections

An adaptation of S2eigv.m, S3eigv.m allows the user to choose three eigenvalues. The function plots the data according to the corresponding eigenvalues, and eigenvectors. An example of a 3-Dimensional Eigenvector Projection can be found at the bottom of page 30.

3.1.3 3-D Fisher Projection (All Classes)

An adaptation of S2fshpac.m, S3fshpac.m allows the user to select three of the calculated eigenvalues. The function then uses all of the classes to calculate the Fisher Discriminant, and plots the data accordingly. An example of a 3-

Dimensional Fisher Projection using all of the classes can be found at the top of page 31.

3.1.4 3-D Fisher Projection (Three Class Pairs)

S3fshp3c.m was adapted from S2fshp2c. It allows you to first select three class pairs, and then select one of the calculated eigenvalues for each class pair. The function then calculates the Fisher Discriminant according to the three eigenvalues, and plots the data. An example of a 3-Dimensional Fisher Projection can be found at the bottom of page 31.

3.2 Feature Evaluation

The newly added feature evaluation functions are adapted from an external Inter-distance to Intra-distance Measurement Ratio (IIMR) Program developed by R. M. Floyd. They include 3 metric measurements, as well as a process comparison tool. Many of the IIMR files were added to STATPACK in order to keep a similar interface. These functions allow a user to edit plots that are displayed, as well as to print them to the clipboard or a printer.

The metric tools can calculate a distance both globally as well as pair-wise. They also calculate a feature relative worth measurement for both global and class-pair situations. After doing the calculations, the metric tools output a *.mat file into the node's folder, that can be used by future calls to that metric and the process comparison function. This process comparison takes user specified files, and compares them to determine which gives the best separation between classes.

These functions can be accessed under the "Feature Evaluation" menu at the STATPACK main screen. The different display options can then be selected under the internal "menu" menu. Appendix C has example Global Feature Relative Worth plots for the feature evaluation options, as well as some Process Comparison charts.

3.2.1 IIMR

The Inter-distance to Intra-distance Measurement Ratio is the first of the feature evaluation metrics. It has three options: Original features, Decorrelated features (using a pooled covariance matrix), and Decorrelated features (using class-wise covariance matrices).

$$IIMR_{jk} = \frac{1}{p}\sum_{l=1}^{p}\frac{b_{ljk}^2}{w_{lj}^2 + w_{lk}^2}$$

$$FRW_{ljk} = \frac{b_{ljk}^2}{p \cdot IIMR_{jk} \cdot (w_{lj}^2 + w_{lk}^2)}$$

Here, classes j and k are some class pair, p is the number of features and l is a feature of interest. "b^2_{ljk}" is the square distance between the means of feature l for classes j and k. "w_{lj}" and "w_{lk}" are the variances of feature l for classes j and k respectively.

The first option does the calculations according to the current features and distances. The second and third options show what the calculations would be if

the user was to run the feature decorrelator (described in paragraph 3.3) for either of it's options.

An Example of a Global Feature Relative Worth Plot for each option is on page 33.

3.2.2 Mahalanobis Distance

The Mahalanobis Distance (MD) option has three options: Size, Shape, and Total Distance. Each option calculates the distance according to the corresponding section of the equation. It also calculates the feature relative worth by leaving a single feature out of the calculation, and measuring the difference between that distance and the original one. From [4] we have:

$$\delta = \mu_j - \mu_k$$

$$MD_{jk} = \frac{1}{2} tr[(C_j - C_k) * (C_k^{-1} - C_j^{-1})] + \frac{1}{2}[\delta^T (C_j^{-1} + C_k^{-1})\delta]$$

Shape Size

Here "C_j" and "C_k" are the covariance matrices of classes j and k respectively, and δ is the difference between the mean vectors.

Mahalanobis Global Feature Relative Worth plots is located on page 34.

Note that herein, the sum of the size and shape terms is being referred to as the Mahalanobis Distance. The stricter terminology is as given in [4]. The corresponding equation is therein denoted as equation (40), and can be found on page 1449.

3.2.3 Bhattacharyya Distance

The Bhattacharyya Distance (BD) metric has the same options as the Mahalanobis metric, and they work the same way. In fact, the two measurements are very similar. The main difference between the two, in terms of the size component, is when you take the inverses of the matrices. In the Mahalanobis case, the inverses are taken, and then they are added. However, in the Bhattacharyya case, the adding occurs before the inverse is taken. From [4] we have:

$$\delta = \mu_j - \mu_k$$

$$BD_{jk} = \frac{1}{2}\ln\frac{\left|\frac{C_j + C_k}{2}\right|}{|C_j|^{1/2} * |C_k|^{1/2}} + \frac{1}{8} * (\mu_j - \mu_k)^T * \frac{(C_j + C_k)^{-1}}{2} * (\mu_j - \mu_k)$$

Shape Size

Here "C_j" and "C_k" are the covariance matrices of classes j and k respectively, and μ is the respective class's mean vector.

Please refer to page 35 for example of Bhattacharyya Global Feature Relative Worth Plots.

The Bhattacharyya shape term differs in form relative to the Mahalanobis distance. In [4], the BD equation is referred to as equation 64 on page 1451.

NOTE: All of the above metric functions are contained in feateval.m, and are selected by sending the function the correct string in the calling command.

3.2.4 Process Comparison

The Process Comparison tool (proccomp.m) can be used to assess different ways to collect certain data. It can use the IIMR, the Mahalanobis Distance, or the Bhattacharyya Distance metrics to accomplish this. However, it runs under the restrictions that all data files compared must use the same number of features as well as the same class tags.

To run a comparison, a user must first read all of the desired files into STATPACK. Each selection under the Process Comparison option is the metric that it will use to compare the processes. For IIMR, proccomp will use the option that was selected. Under Mahalanobis and Bhattacharyya distances, it will use the selected part(s) of the equations . If the desired metric has not been calculated on a certain file, proccomp will calculate it. If the metric has been previously calculated, then the function will load the appropriate *.mat file under the node of the selected file.

Once proccomp.m is run, the user can compare the process files in global or class-pair situations. On the plots, positive values reveal an increase in the metric, negative values reveal a decrease, and values of 0 indicate that no change occurred. The red marks on the global plot (as on the metric plots) correspond to the range of class-pair values for that particular comparison.

Proccomp.m has two methods to solve for a global comparison:

$$C_{global} = 10 * \log_{10}(\frac{Global_2}{Global_1})$$

$$C_{global} = 10 * \log_{10}(mean(\frac{dist_2}{dist_1}))$$

Where $Global_1$ and $Global_2$ are the averages of pair-wise distances for processes 1 and 2 respectively. Similarly, $dist_1$ and $dist_2$ are the distances between a given class pair for processes 1 and 2. The mean is used to average over all possible pairs. Thus the first method forms a ratio of averages and the second forms an average of ratios.

It can also solve class-pair comparisons using:

$$C = 10 * \log_{10}(\frac{dist_1}{dist_2})$$

Examples of Global Comparison Plots can be found on page C-5.

3.2.5 New *.mat Files

In order to connect the metric functions to the process comparison, the use of interim *.mat files was necessary. These new files are located within the selected node, and contain the metric data. This data includes the class-pair metric matrix (tdist – # class-pairs x 1), the global metric value (tGdist – 1 x 1), and the list of class-pairs (Plist - # class-pairs x 2). The file names are Iimr.mat, Pimr.mat, Dimr.mat, Msize.mat, Mshap.mat, Mdist.mat, Bsize.mat, Bshap.mat, and Bdist.mat. Each file contains the information for the corresponding metric (Mahalanobis Size data would be in Msize.mat). The decorrelated IIMR files are in Pimr.mat for a pooled covariance matrix, and Dimr.mat for individual covariance matrices.

The process comparison function and the metrics can then read these files. That way, the metrics need not be calculated every time a feature evaluation option is called. This can save a lot of time, especially with larger *.dat files.

3.3 Feature Decorrelator

The next recent addition is the data decorrelator. The discussed operations are run by calling decorr.m. The function call is under the "File" menu on the STATPACK main screen. This function will output a *.dat file with decorrelated data from the current node. This process can be done one of two ways: using class-wise covariance matrices, or using a pooled covariance matrix.

Note: The user can preview possible results through the IIMR function.

Class-wise Covariance Matrices:

$$X_{id} = X_{io} * eig(\text{cov}(X_{io}))$$

Pooled Covariance Matrix:

$$X_{ad} = X_{ao} * eig(\frac{\sum \text{cov}(X_{io}) * nvc}{nv - nc})$$

```
Where:
nvc-# vect/class
nv-# vectors
nc-# classes
a-all classes
i-class
d-decorrelated
o-original.
```

Both options have practical uses. Using class-wise covariance matrices results in a very large visual separation between classes when the data is plotted. The problem is that if the user is attempting to classify unknown data, they may not know which covariance matrix to apply to the unknown data. Using a pooled covariance matrix results in poorer separation. However, since there is only one covariance matrix, it is easier to apply to an unknown data set.

During the file out, the program reads in the current file name, and creates the output file name accordingly. If the file in use is not already decorrelated, the output file will have a name of "dc1" then the first five characters of the current file's name. If it is already a decorrelated data set, the program will increment the number in the third spot in the name. Decorrelating dc1nasa.dat would result in a file named dc2nasa.dat. However, this will only proceed until it number becomes a "9". At that point, it will increment to a "0", and then start over writing over any existing file with the same name.

The written *.dat file is placed in the data directory under the STATPACK folder. After the decorrelated data set has been created, it can be loaded into STATPACK, and viewed as any other *.dat file.

3.4 Data Set Generator

The final new feature is the Random Data Set Generator. This function is found in datagen.m. The function call for this routine is also under the "File" menu on the STATPACK main screen. Through this function, the user can

specify the output file name, number of classes, number of features, and number of vectors per class (must be greater than 5 times the number of features) that will be created in the output *.dat file. Next, they can specify the means for each class and feature, or allow them to be generated randomly. The user must then specify whether each class is to have it's own covariance matrix, or if there is to be a single covariance matrix that will apply to all classes. The next question is whether or not the user wants to specify the covariance matrices. If not, they will be generated randomly by multiplying the randn() output by 10. Otherwise, the user must input an upper triangular matrix of values to use for each desired covariance matrix. Then, using the symmetry of the covariance matrix, the function will create the user defined covariance matrices. If the user is creating individual covariance matrices, they must also input values for the feature means. If they are creating a global covariance matrix, the mean values will be generated randomly. The mean values are placed into a matrix such that each row of the mean matrix contains the corresponding vector's intended mean values for each feature. The function will then take the covariance matrices and means, and apply them to randomly generated data.

The User may then choose to specify feature names, in the same manner as all other inputted information. If the feature names are not specified, STATPACK will apply the default names. The output file is then saved in the data directory under the STATPACK main folder.

The Data Generator also has one other capability. If a user wanted to specify individual covariance matrices, they would be forced to type in a possibly very large amount of numbers (up to 21,700). And, they would have to do this every time they wanted to change even a single number. To alleviate this problem, a save/load feature has been established.

The matrices were saved in *.mat file format, so that the user could change any number easily from the Matlab command line. The *.mat files are named "covm"-incremented number-".mat", and the matrix is saved in the variable covmatrix. The relevant equation is:

$$C_i = (randn(nvpc, nf) * evc(CM)^{-1} * evl(CM)) + \mu l$$

Where:
μl - mean list
nvpc - #vec. / class
nf - # features
CM - Cov. Matrix
randn - rand. # mat.
C - Class
evc - eigvector mat.
i - class number
evl - eigvalue mat.

4.0 Testing

In order to test the new version of STATPACK under varying conditions, five test *.dat files were created using the new data generator feature. All files had 5 classes, and ten features. The global covariance matrix for each file was a diagonal matrix with ones on the main diagonal, and zeros off of it. However, the means were changed in ways such that different situations could be experienced:

- datagen1.dat – only one feature's means separated the classes
- datagen2.dat – the feature means had increasing separation (feature 1 separated classes by 1, feature 2 separated classes by 2, etc…)
- datagen3.dat – only two feature's means separated the classes, and they did it equally
- datagen4.dat – all of the feature means separated the classes equally
- datagen5.dat – all of the feature means were equal, i.e. no separation in the means

These files were then run through the feature evaluation metrics and the analysis plots to determine if everything calculated as it should have. All of the analysis plots came out as expected, which not only proves that they are correct, but it also proves that the data generator is acting correctly. The feature evaluation metrics had no trouble on the first three files. However, when the Global Feature Relative Worth was calculated on the fourth and fifth files, the results were at first surprising. In stead of the bar graphs being straight across, the values varied. Then, it was realized that there was still variance, which could lead to differences between the features. These differences would be slight, but since the feature worths are relative, the result is a graph with larger than expected variance in itself. Therefore, it has been determined that the previously described portions of the program can be trusted to display accurate results. Plots to illustrate the testing process are located in appendix D. These include two feature projections to visually illustrate the difference in the separation that each feature can give, as well as a GFRW plot to analytically display the separation that each feature is capable of applying.

5.0 Future Development

There are certain portions of STATPACK that still require further attention. These include both classifier functions and the subnode setup. The following is a summary of the remaining work that has yet to be done.

5.1 Nearest Mean Classifier (nmclass.m)

The Nearest Mean Classifier attempts to classify data based upon which class's mean vector the vector in question is nearest to. It uses the standard deviations of each class and feature as well as a built-in metric calculator to accomplish this.

Although certain pieces now work, that is not true for all options in all cases. Currently, the full-set / self-classify option works completely. However, the average-set / self-classify only works when the percentage of the data set is >=50%. Also, the comp.-classify options have errors at both ends of the percentage scale. These errors need to be corrected in a way that still gives accurate results without removing options from the function.

5.2 Fisher Classifier (fisherc.m)

The Fisher Classifier function uses the fisher discriminant to try to classify the data.

This function is named "fisherc.m". It was started by S.P.Montana during the summer of 1998, and was never completed. Currently, it can not be accessed through STATPACK, but the *.m file is in the "Classify" folder of the

STATPACK directory tree. It needs some time and effort to be understood, and finished so that it can be implemented. If added to STATPACK, it should prove itself to be a useful tool.

5.3 Subnodes

Currently, if the user selects a subnode instead of a main node, STATPACK will encounter many various errors while attempting ordinary operations. These errors are not localized to a certain set of functions either. A serious effort needs to be put into this so that STATPACK can be utilized fully.

6.0 Conclusion

The recent changes in STAPACK add to the utility of an already useful program. With the addition of 3-Dimensional Plotting, the user is able to visually find a much higher seperability than they may using only 1 or 2-D plotting functions. With the new feature evaluation functions, the user can tell which features will give them that high separability.

Also, STATPACK has been made compatible with Matlab version 4.2. Although its appearance is different, and it is not as fast, it still works. This should allow it to be more useful, as it can be ported to more machines that may not have version 5.3 available to them. Now, with the release of version 6, STATPACK may be able to become faster and have even more utility as the Matlab functions become cleaner and more powerful.

7.0 Acknowledgements

The author would sincerely like to thank Dr. Andrew Noga for his mentoring and assistance during the period discussed. He would also like to thank Rich Floyd as well as the other members of the AFRL that he has come into contact with throughout the summer for the help that they have given. He is very appreciative of the opportunity given to him by the AFRL Summer Engineering Aide Program; through which, he learned more about true engineering than could be gained in any classroom.

[1] -- Shaun P. Montana, "A Statistical Pattern Recognition Tool", In-House Report, RL-TM-96-8.

[2] -- Richard M. Floyd, "TECHNICAL NOTE: A Statistical Pattern Recognition Tool (Modifications)", October 1996.

[3] -- Shaun P. Montana, "Enhancements to a Pattern Recognition Tool", In-House Report, RL-TM-97-4, March 1998.

[4] -- Joseph P. Campbell, Jr., "Speaker Recognition: A Tutorial", 'Proceedings of the IEEE', Vol. 85, No. 9, September 1997

Appendix A

List of Matlab routines for STATPACK

Routine List

The following is a list of unchanged, pre-existing routines written for STATPACK.

bindiv.m
callbstr.m
cdnode.m
closefig.m
clrglb.m
crdtset.m
current.m
delay25.m
delfigs.m
dialogbx.m
editplot.m
fileout.m
fisherc.m
fishvote.m
fj2.m
fsize.m
ha1id.m
ha1menu.m
ha2id.m
ha2menu.m
haself.m
haselfo.m
hahs.m
hamenu.m
hanal2.m
haplot.m
haprnt.m
harange.m
hasel.m
hclass.m
hnode.m
hstpk.m
idclick1.m
idclick2.m
isup.m
mdnode.m
menu4.m
menu5.m
mfeature.m
mlver.m
newclass.m
numtext.m
overlap.m
s1crdv.m
s2proj.m
s2save.m
shlist.m
spmovie.m
textbox.m
threscal.m
time.m
viewdata.m
waitbar2.m

Routine List

The following is a list of STATPACK routines that have been added and/or edited during June 2001 – August 2001.

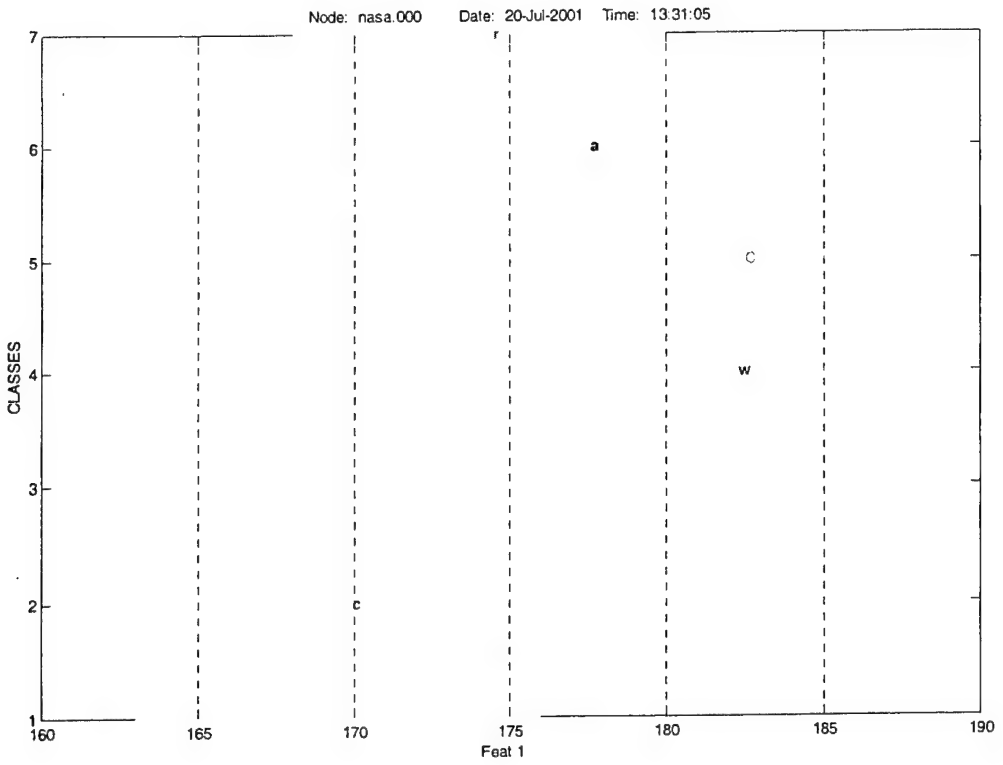
clrlist.m
datagen.m **
decorr.m **
delnode.m
featureval.m **
figaxis.m*
figdtext.m*
figedit.m*
figgrid.m*
fighelp.m*
figjoint.m*
figlabel.m*
figlabel.m*
figload.m*
figmenu.m*
figmouse.m*
figmsave.m*
figmsg.m*
figprint.m*
figptext.m*
figrdrw.m*
figsave.m*
figscrl.m*
figsel.m*
figsela.m*
figsindx.m*
figstart.m*+
figtext.m*
figxypos.m*
figzoom.m*
filein.m
habout.m
hanal1.m
hanal3.m **
hfeat.m **
hfile.m
hide.m
moon.m **
nmclass.m
pathsp.m
pickvec.m
plot1d.m
plot2d.m
plot3d.m **
posfig.m*
proccomp.m**
rffroot.m *
s2crdv.m
s2eigv.m
s2fshp2c.m
s2fshpac
s3crdv.m **
s3eigv.m **
s3fshp3c.m **
s3fshpac.m **
shownode.m
statpack.m
stpkroot.m
subp1d.m

- Pre-existing functions
* - Brought in from IIMR Program
** - Created by B.P.Costello

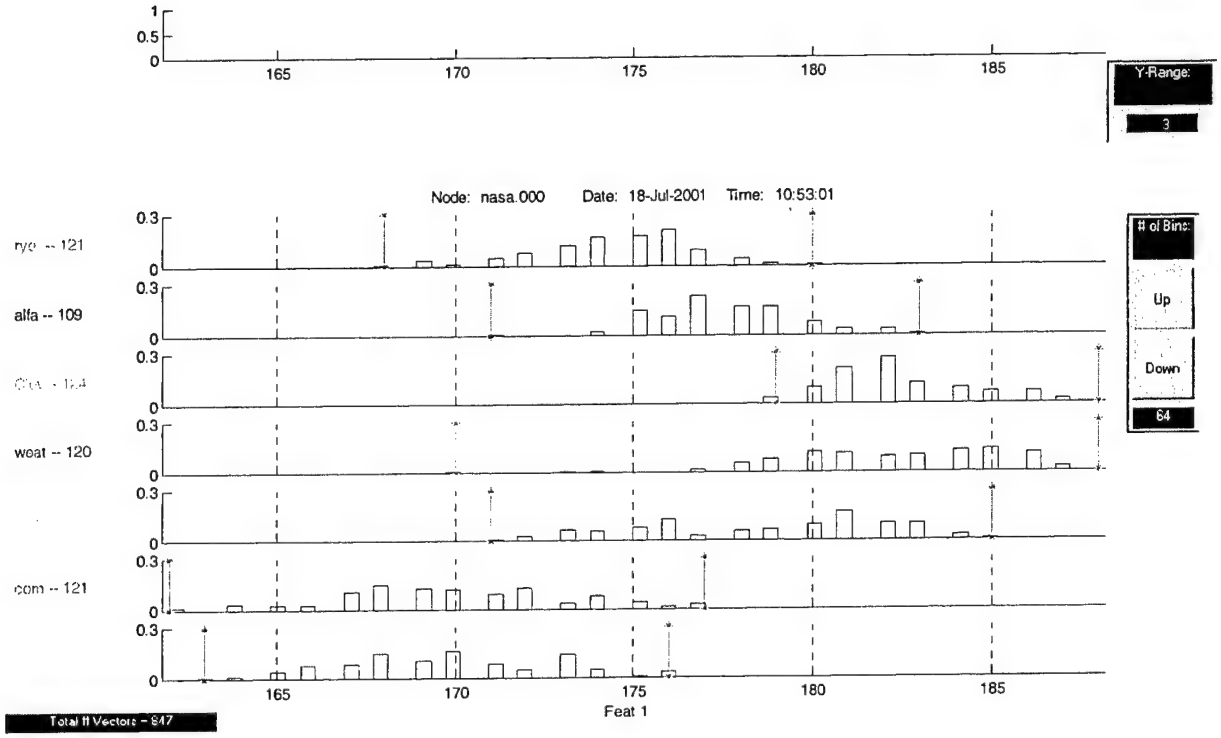
Appendix B

Sample Analysis Plots

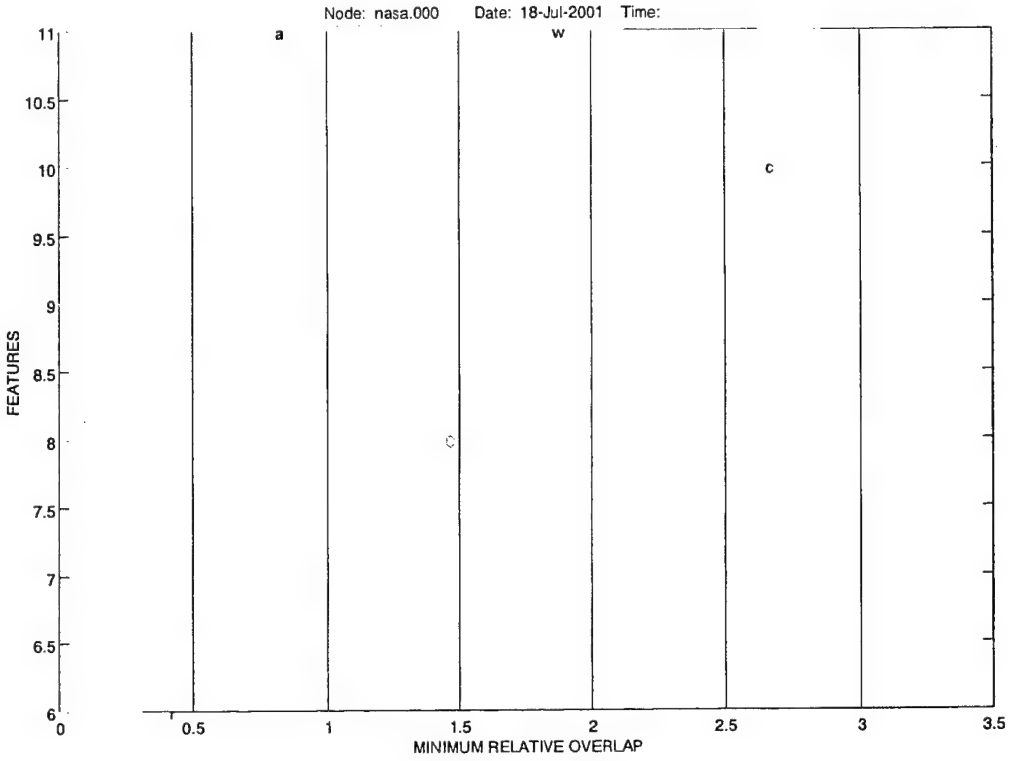
Sample Analysis Plots



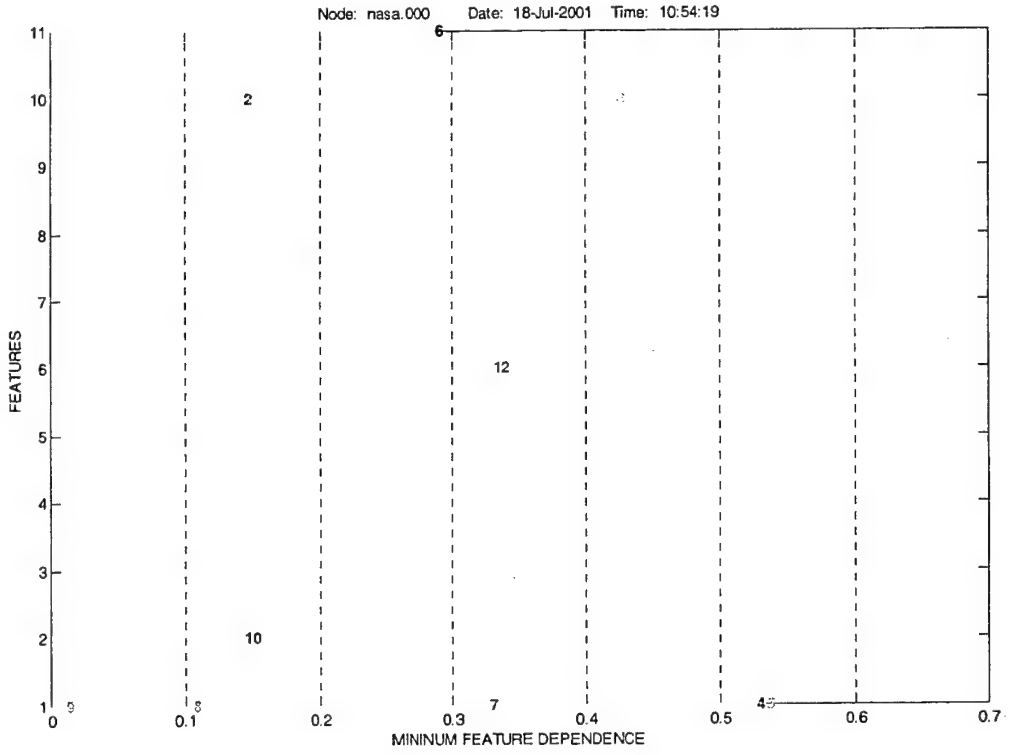
1D Class Ranges ⇑ ⇓ 1D Class Range Intensity



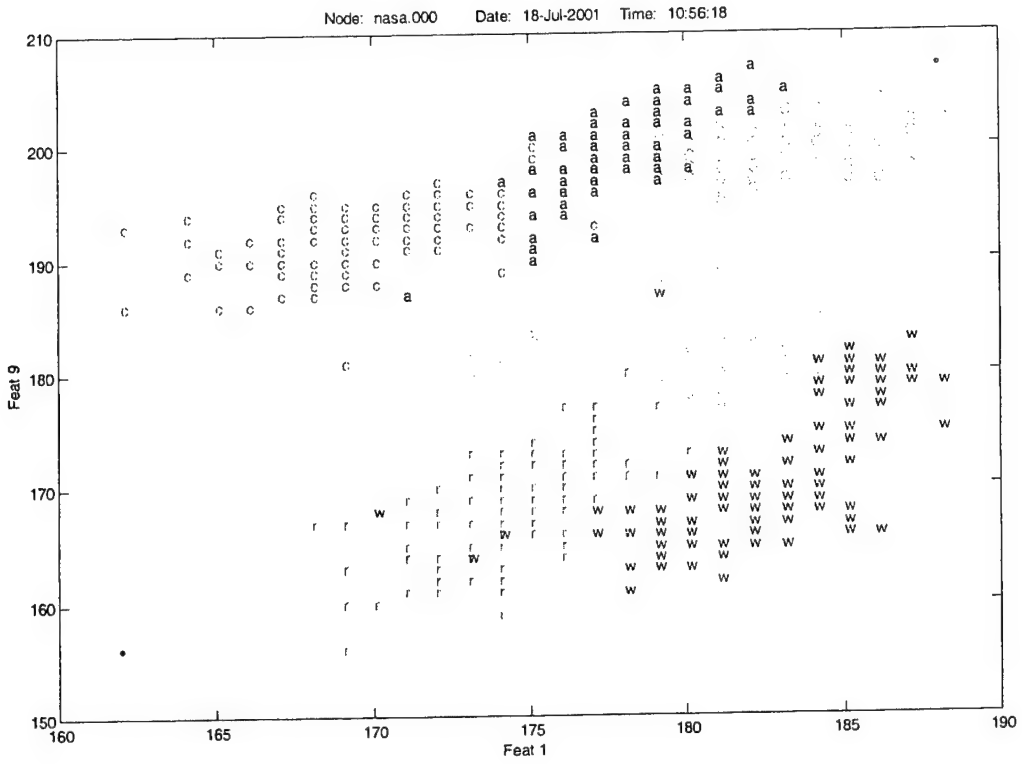
Sample Analysis Plots



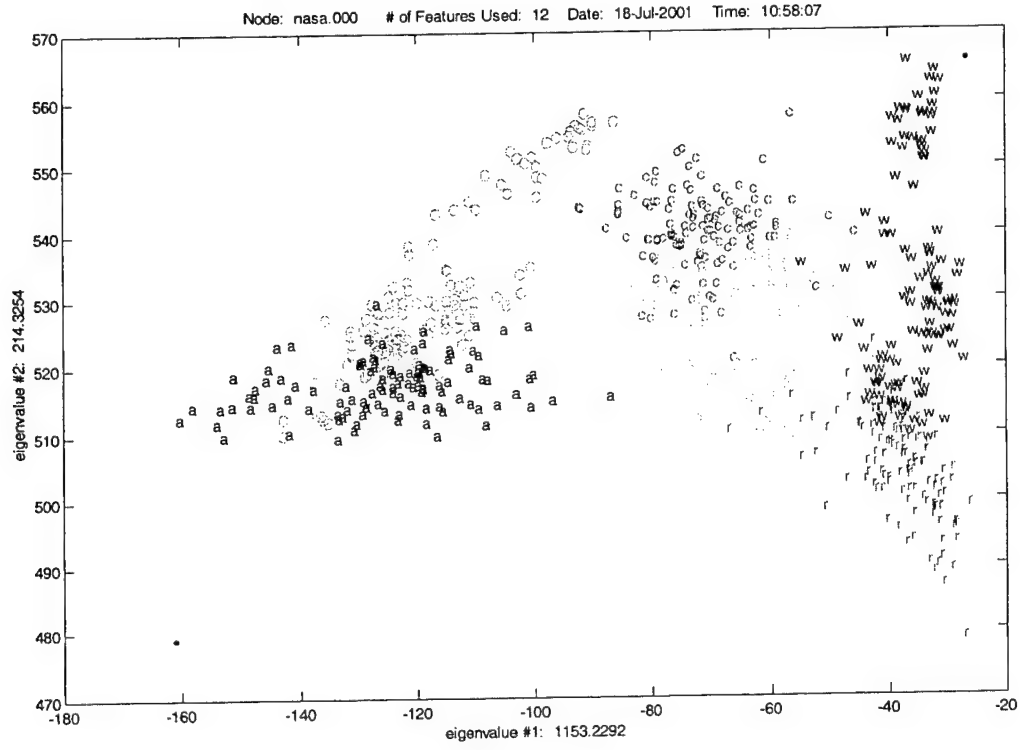
1D Class Overlap ⇑ ⇓ 1D Feature Independence



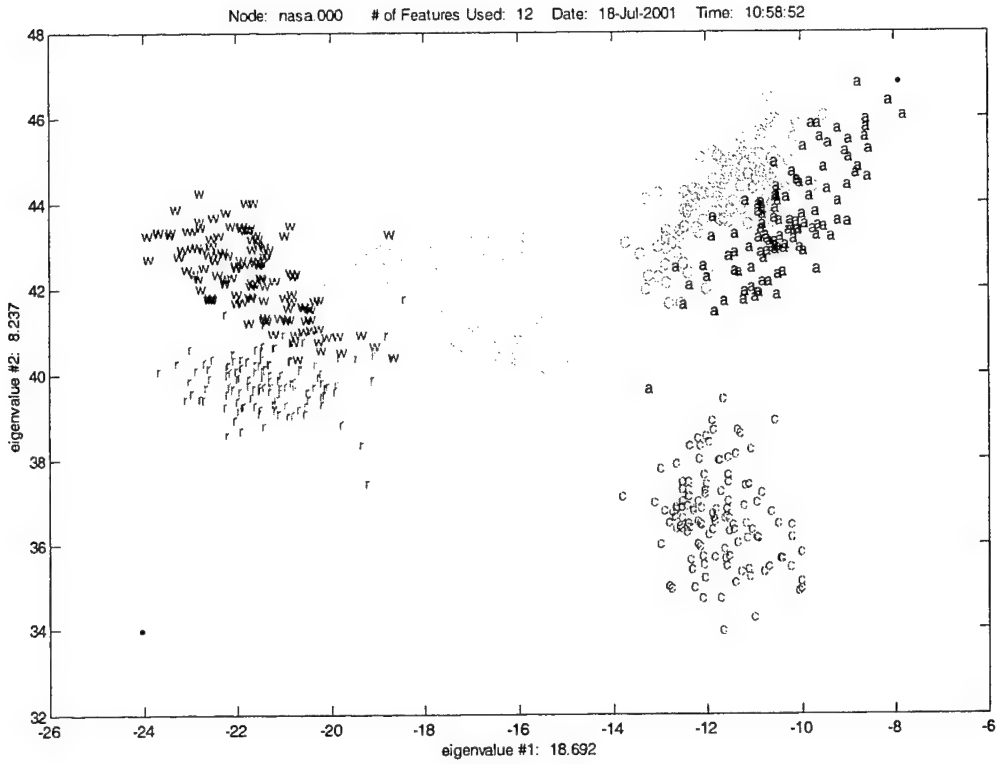
Sample Analysis Plots



2D Feature Projection ⇑ ⇓ 2D Eigenvector Projection

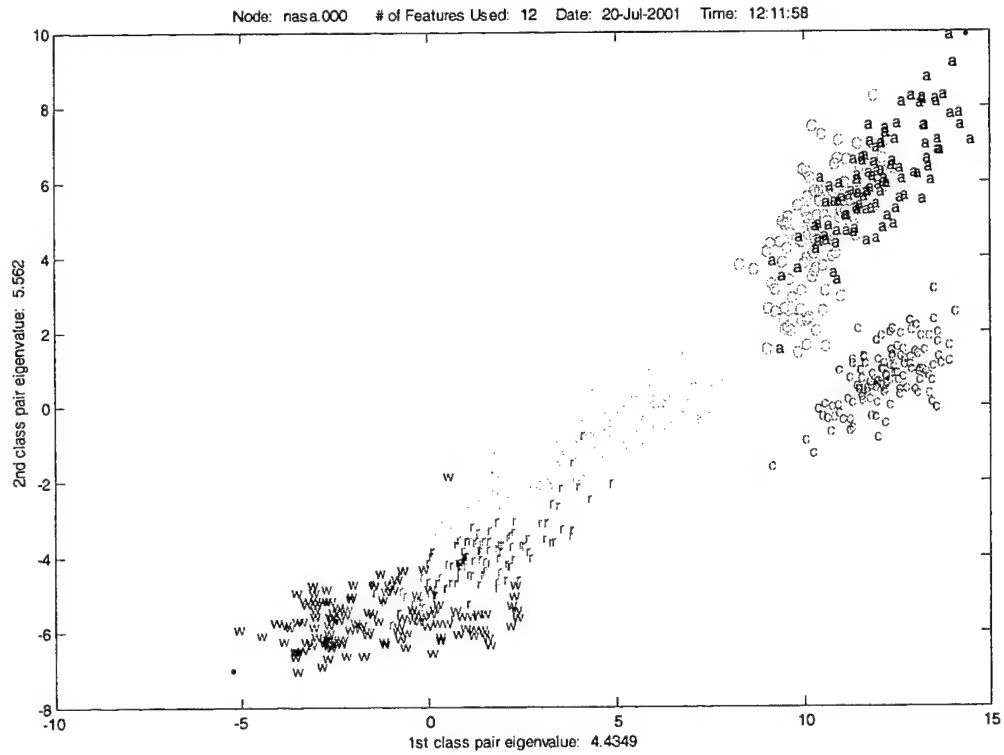


Sample Analysis Plots

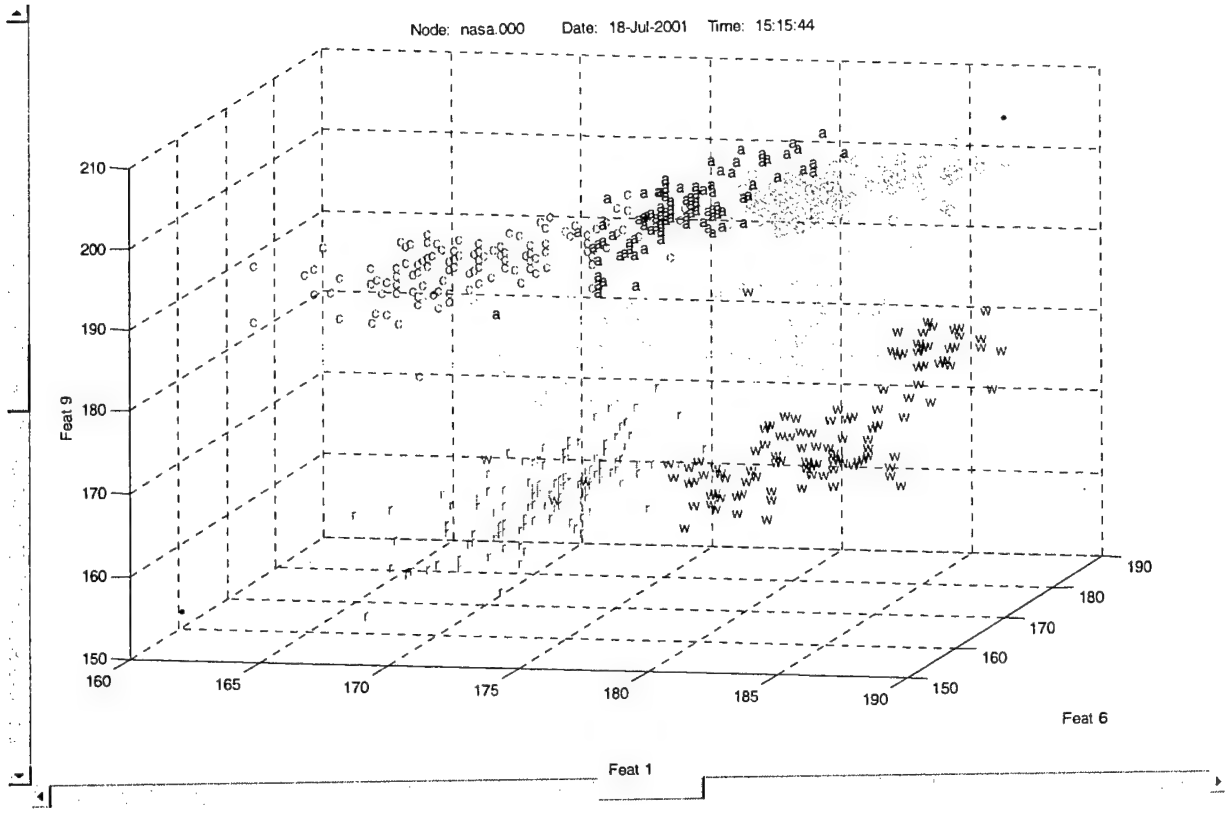


2D Fisher Projection (All Classes) ⇑

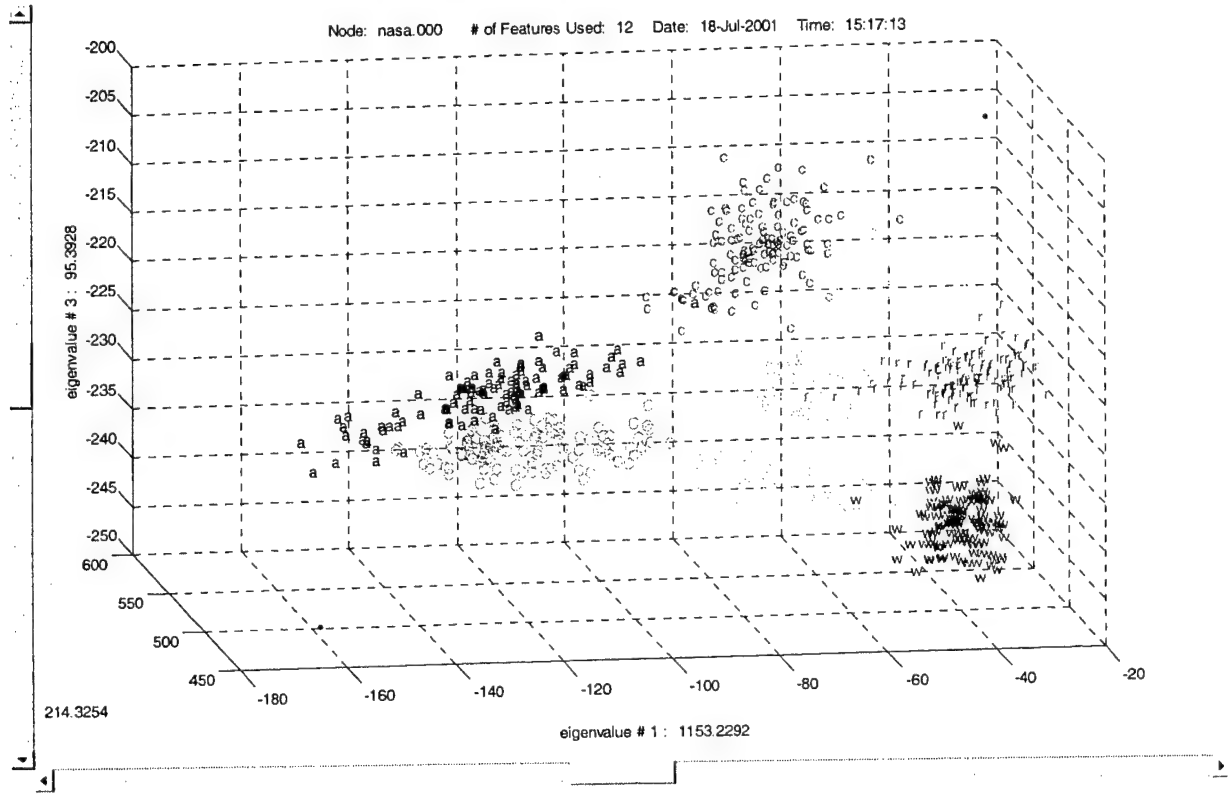
⇓ 2D Fisher Projection (Two Class Pairs)



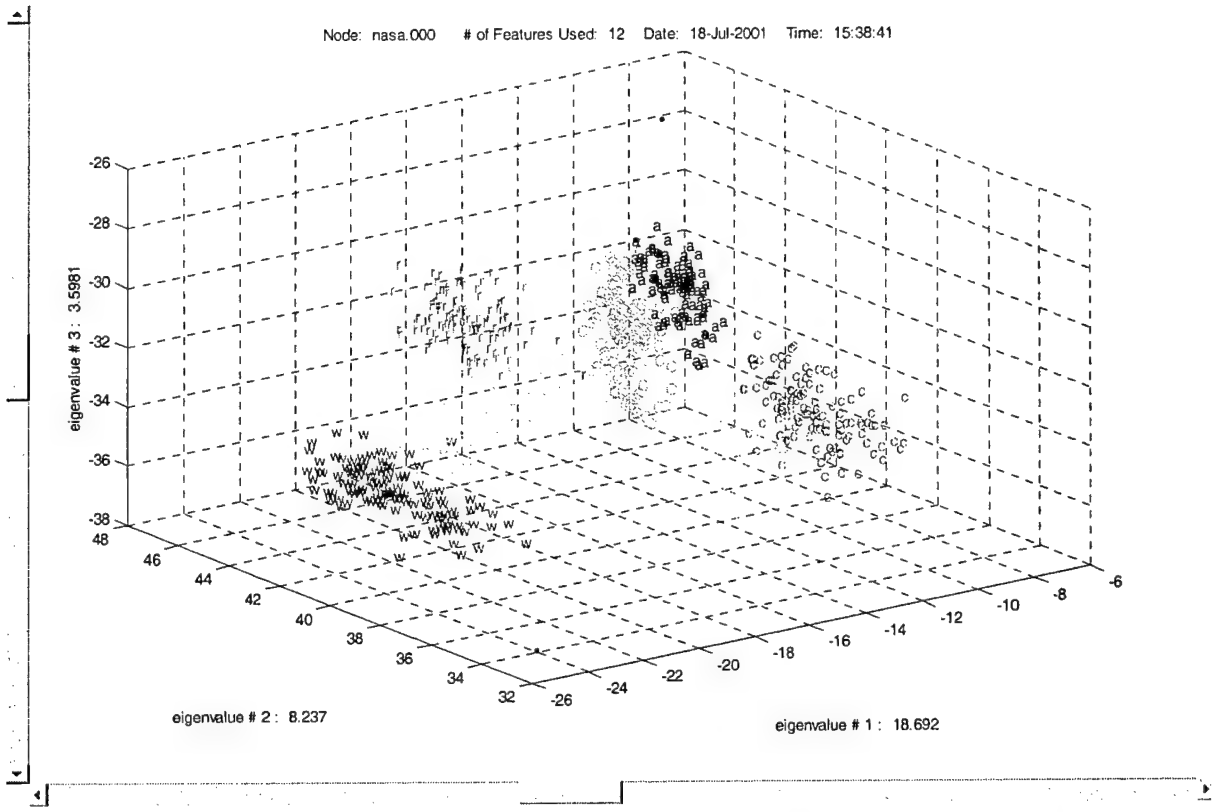
Sample Analysis Plots



3D Feature Projection ⇑ ⇓ 3D Eigenvector Projection

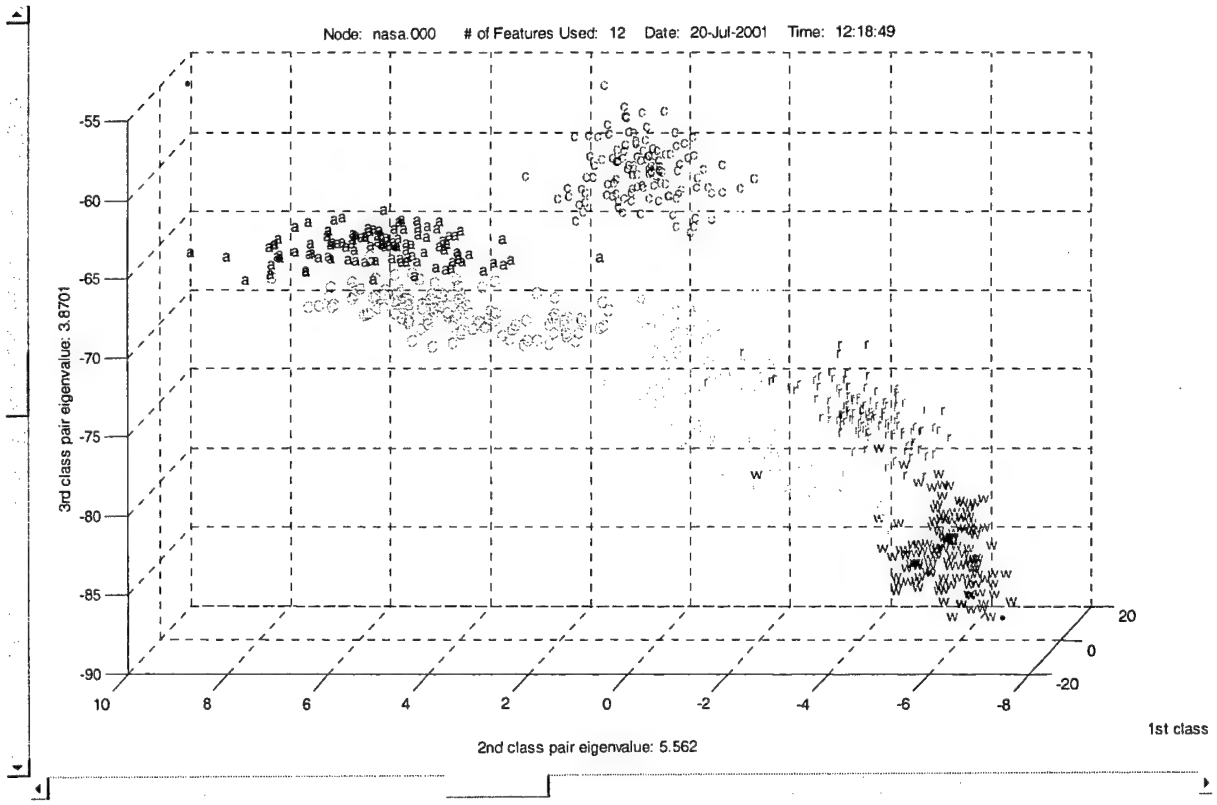


Sample Analysis Plots



3D Fisher Projection (All Classes) ⇑

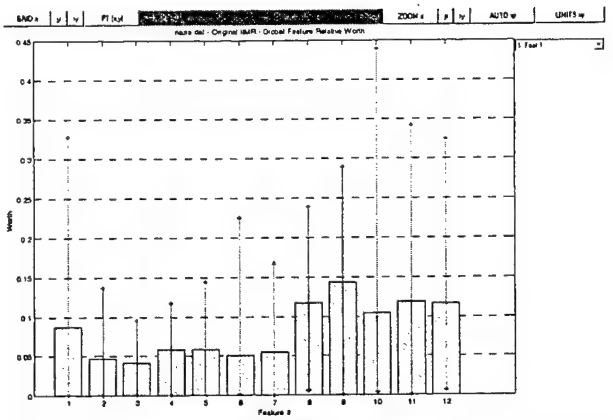
⇓ 3D Fisher Projection (Three Class Pairs)

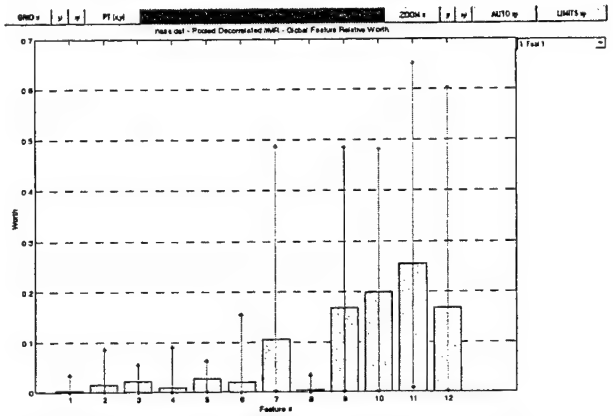


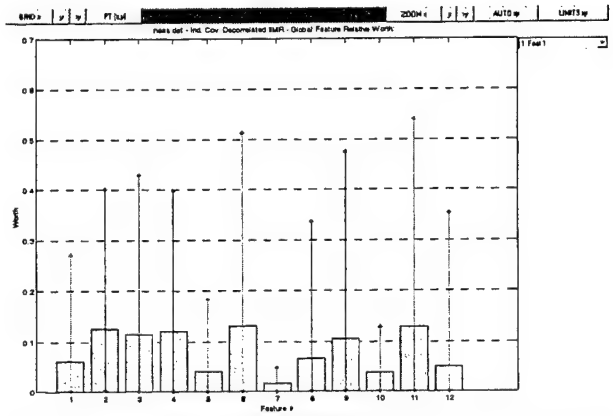
Appendix C

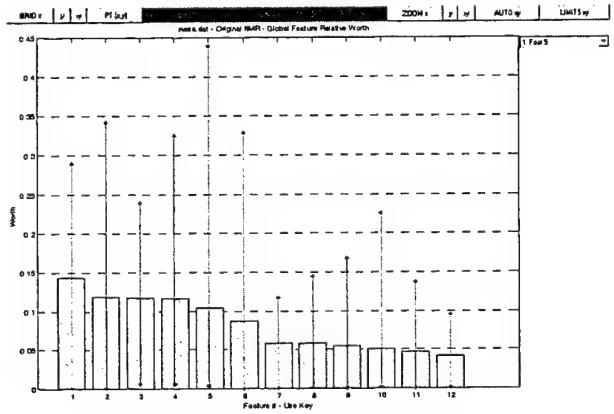
Sample Feature Analysis Plots

Sample Feature Analysis Plots

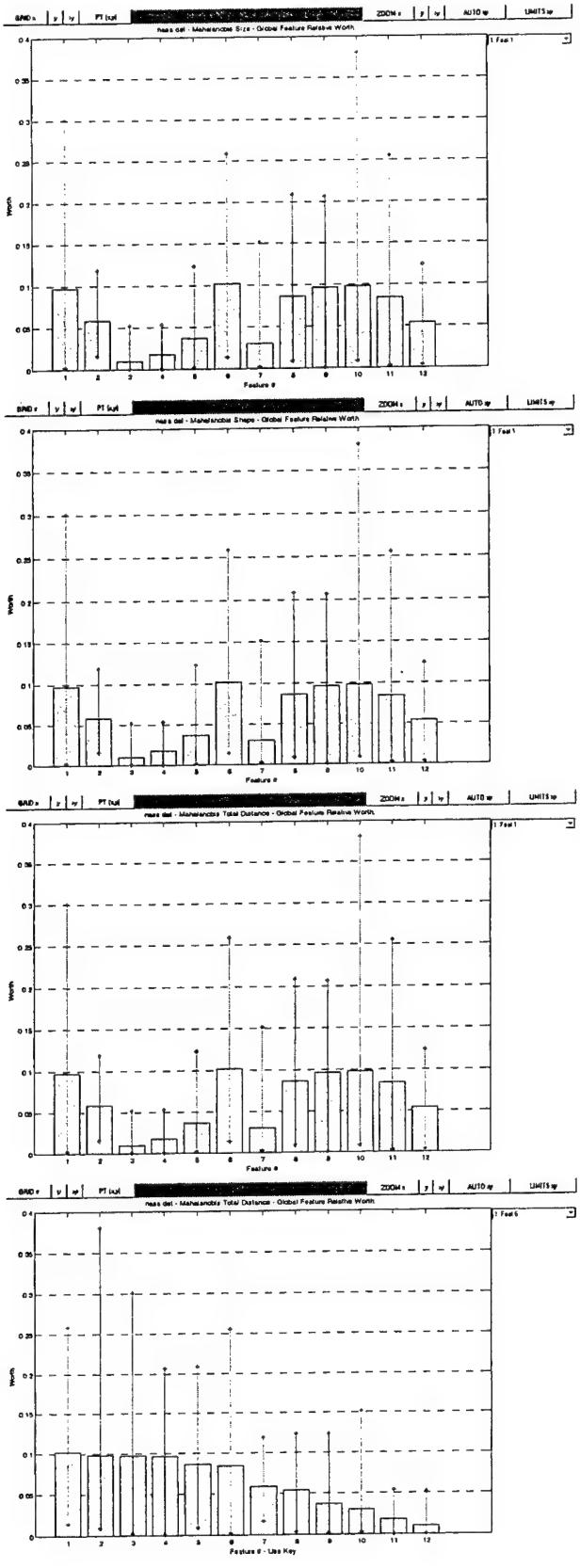




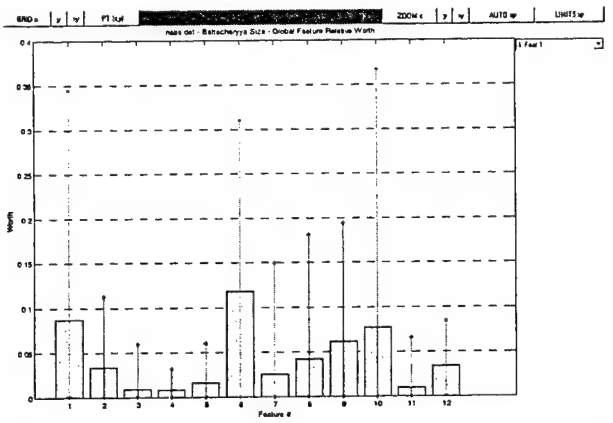


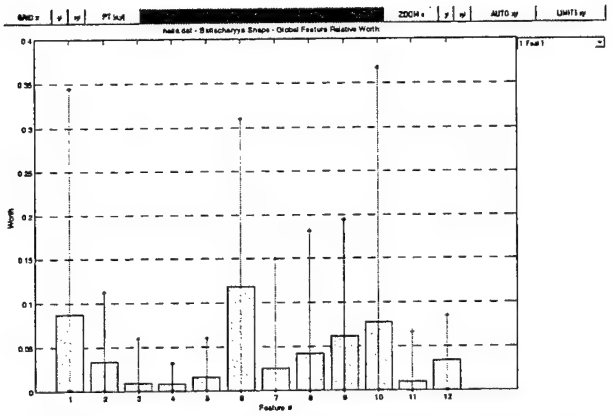


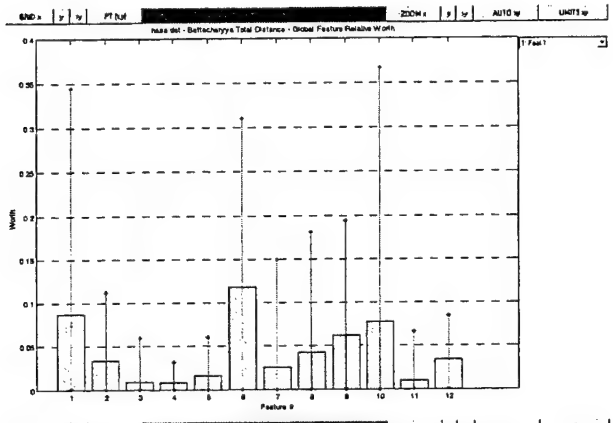
Sample Feature Analysis Plots

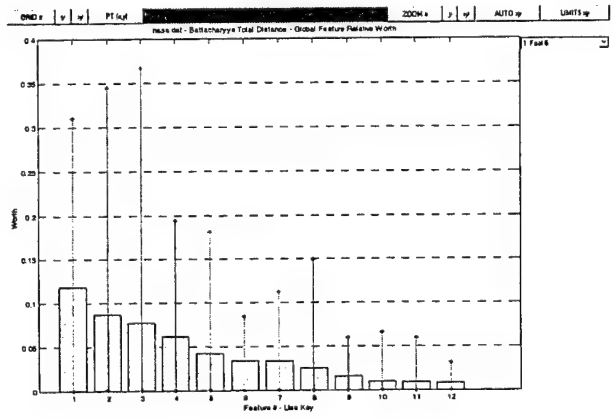


Sample Feature Analysis Plots







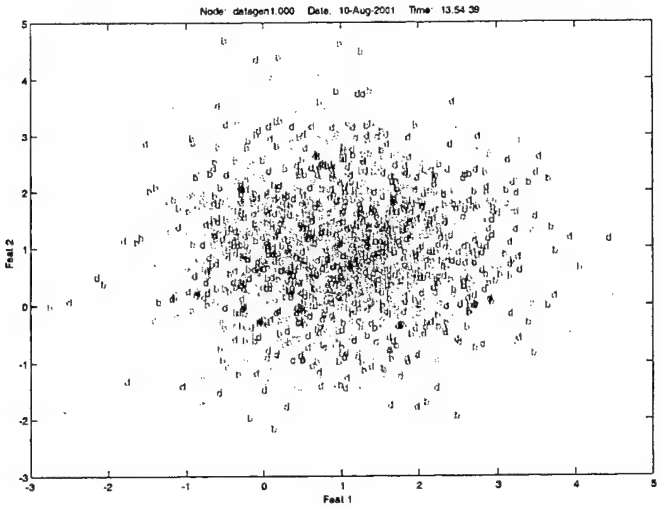


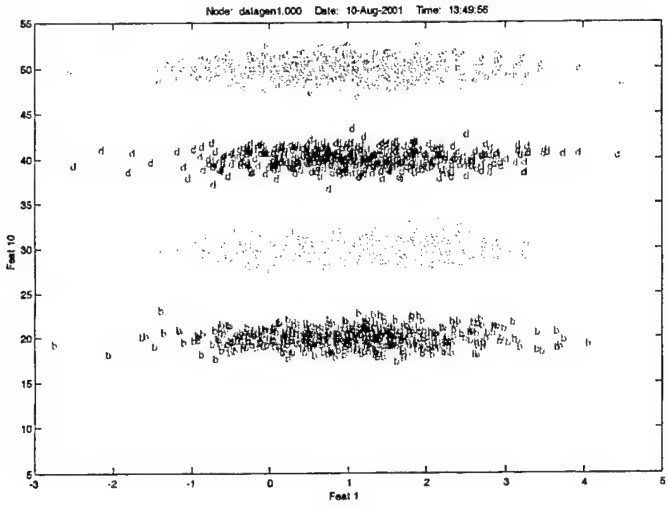
Appendix D

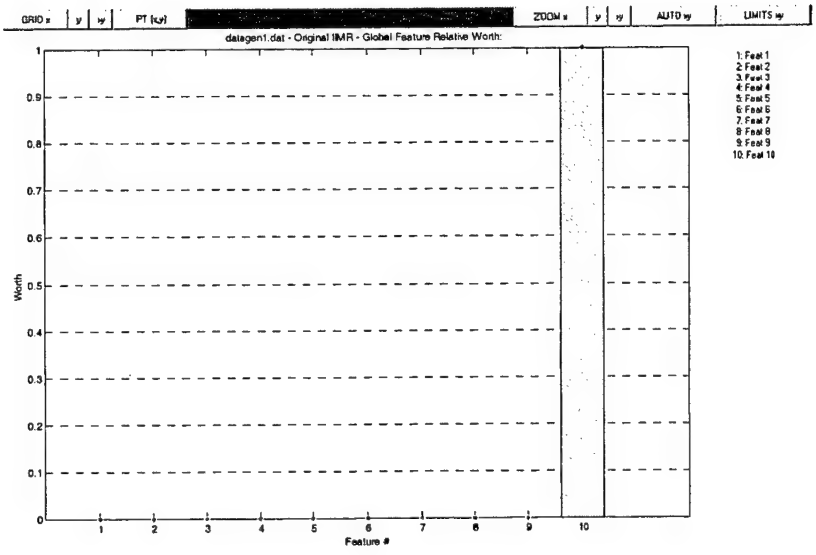
Sample Test Result Plots

Test Result Plots

Datagen1.dat

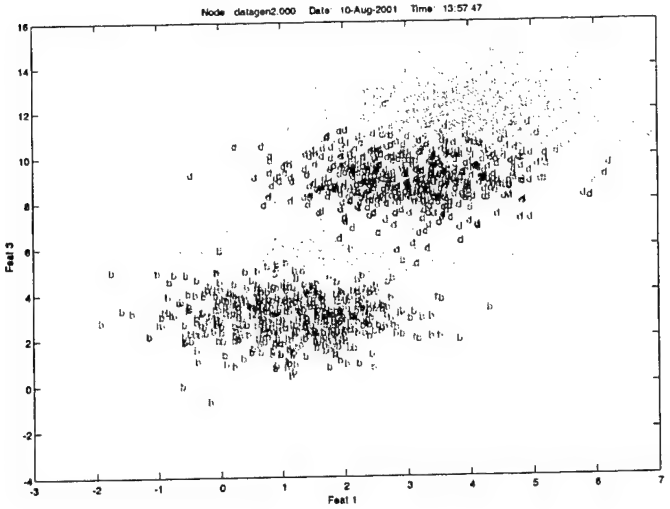


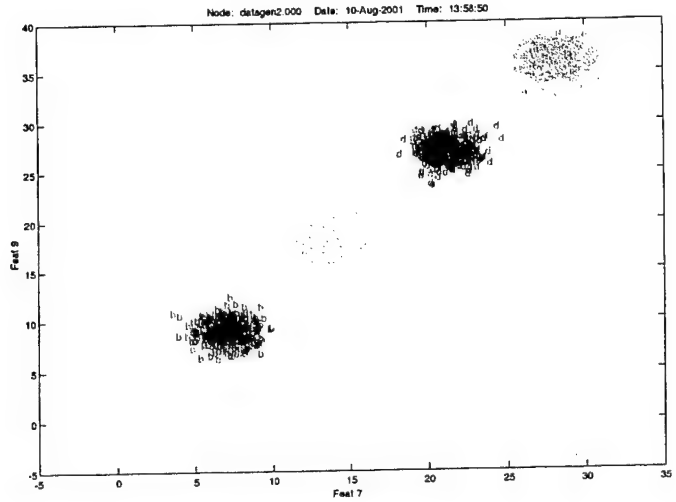


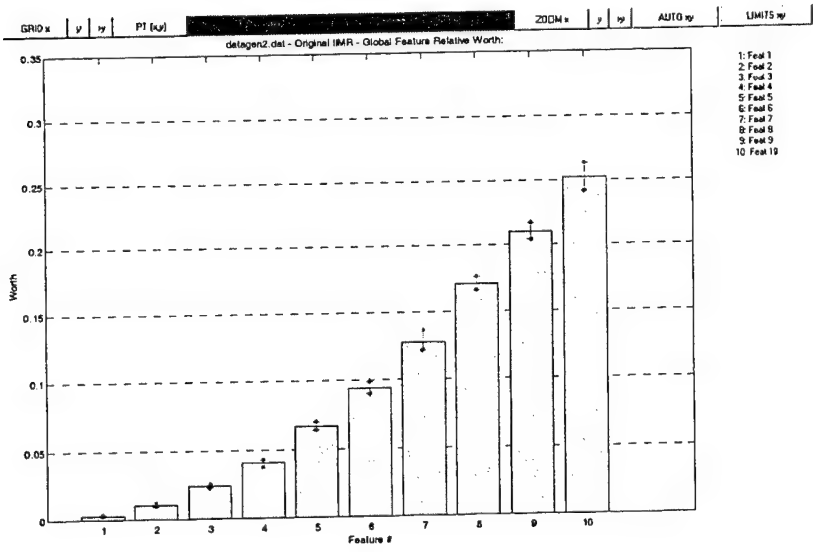


Test Result Plots

Datagen2.dat

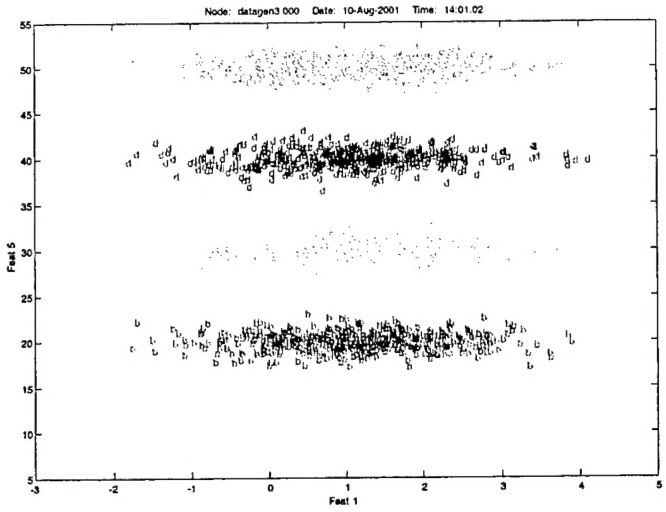


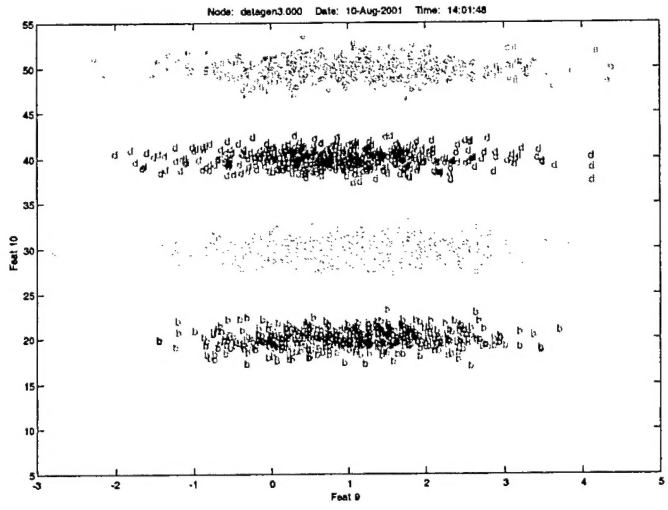


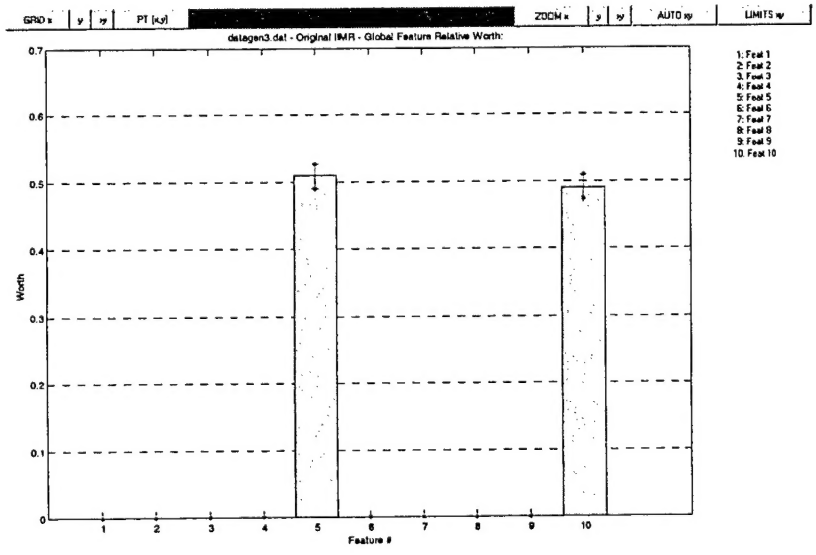


Test Result Plots

Datagen3.dat

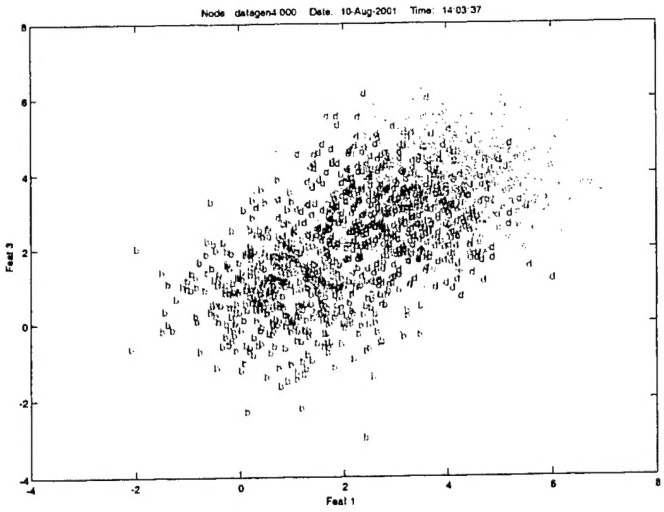


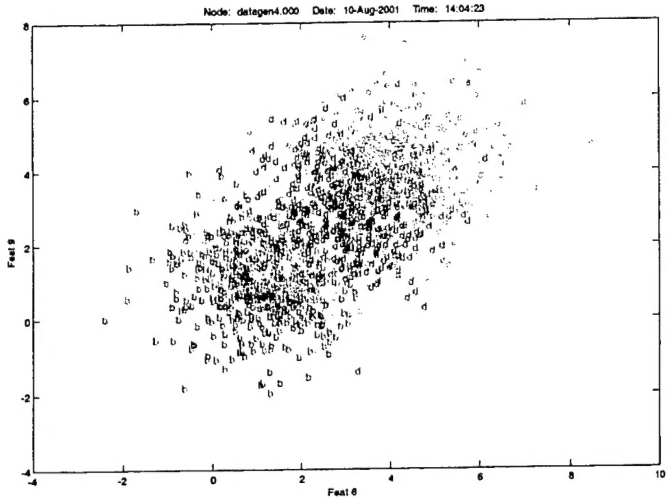


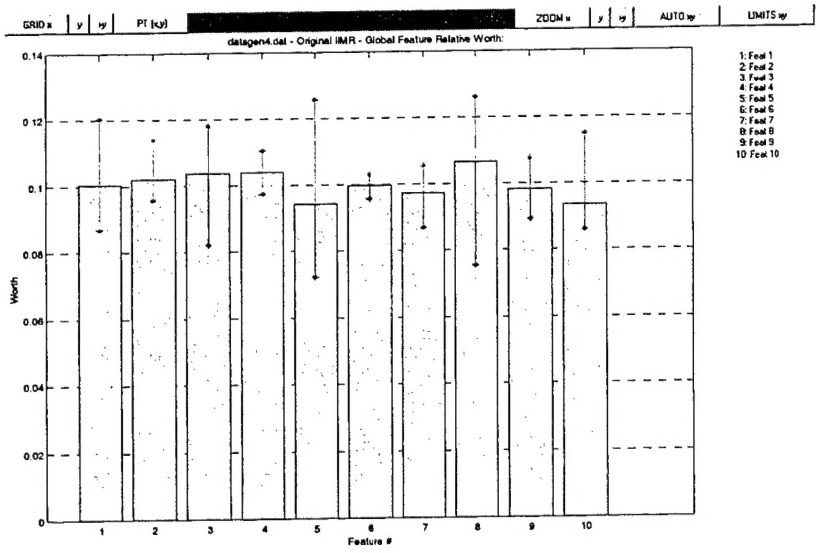


Test Result Plots

Datagen4.dat

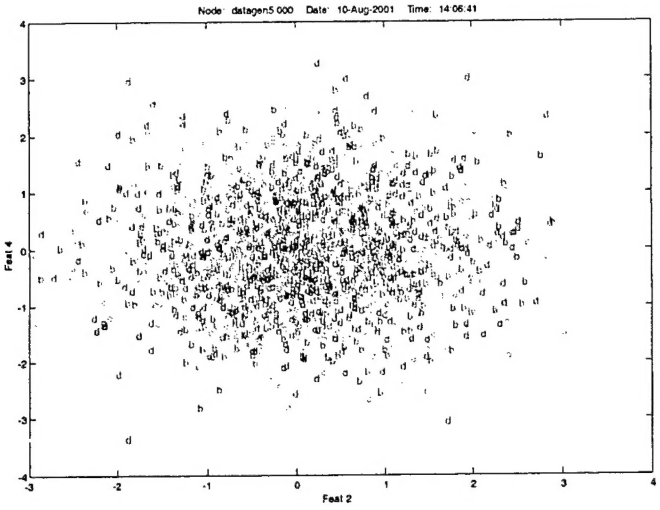


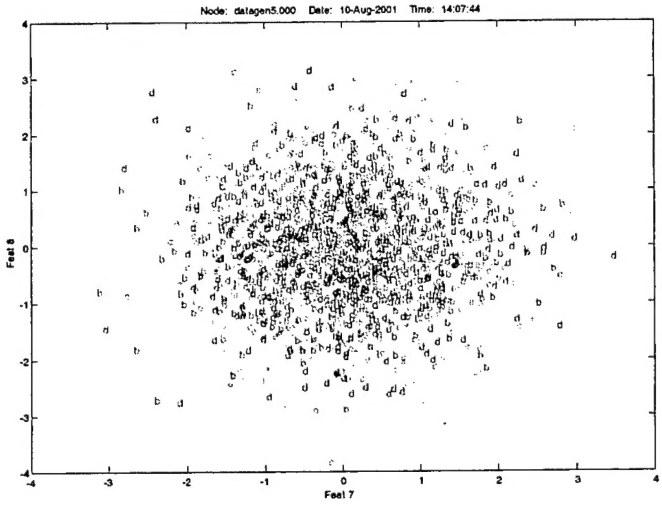


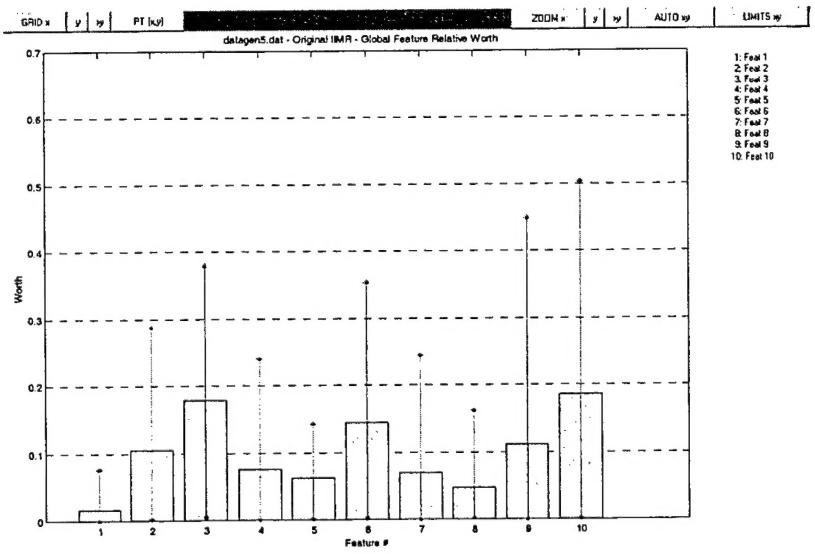


Test Result Plots

Datagen5.dat







MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of Information Systems Science and Technology to meet Air Force unique requirements for Information Dominance and its transition to aerospace systems to meet Air Force needs.